Labs

ATrack AX-7 Setup Guide

Version 1.0

Labs

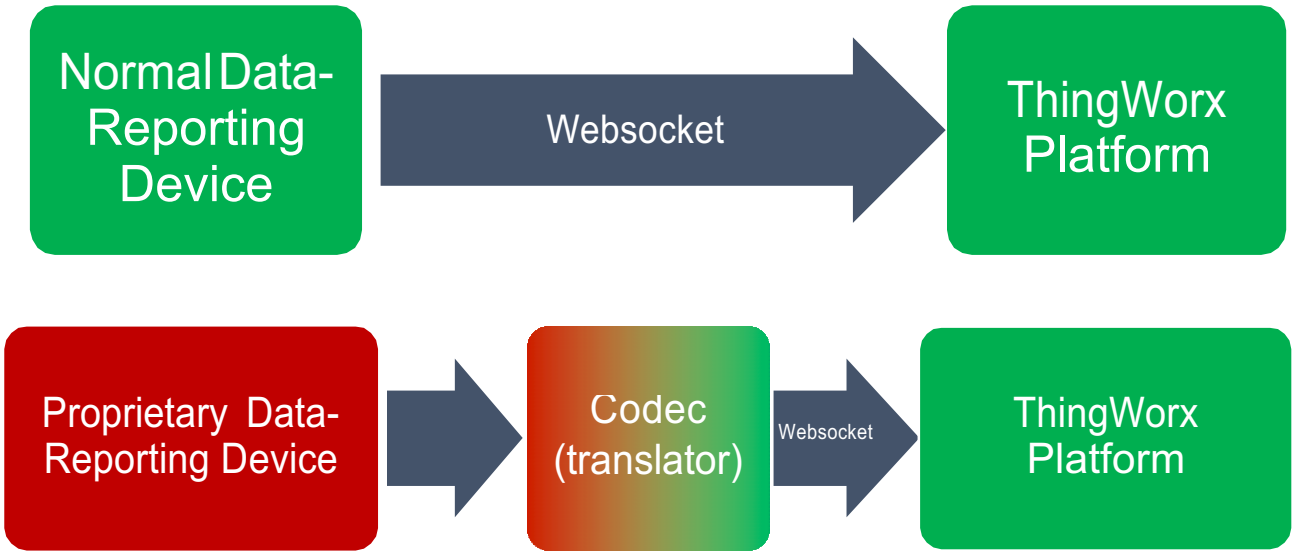
Introduction

About the Protocol Adapter SDK

The Internet of Things is a rapidly-growing and oftentimes fragmented network of devices. Typically these devices do not unify under a standardized "language" that would enable easy connections between them. There are many types of devices that could report their data in a specialized role, but most of these devices would need to report their data in a proprietary format.

In order for the ThingWorx Platform to interpret data from one of these devices, the need for a communications translator, or *Codec*, arises. A Codec operates by transforming incoming messages from a device, and then passing the "ThingWorx-Ready" data to the Platform.

The Protocol Adapter SDK is a codec package provided by ThingWorx for the purposes of connecting such a device, whose firmware and/or data format is unchangeable and incompatible with ThingWorx, to the platform:



Often, these devices open a TCP or UDP networking socket and send data using a proprietary format. To allow these devices to connect to the ThingWorx Platform, the Protocol Adapter SDK runs intermediary between the device and the platform. It listens on a specified port for any incoming device messages, and it parses the incoming messages from the device's native format into properties that are then transmitted to the ThingWorx Platform.

It should be noted that the Protocol Adapter SDK **should ideally be set up to run on an intermediary server**, separate from the one on which the ThingWorx Platform is located (for scalability purposes). However, the Protocol Adapter SDK may be run on the same server on which ThingWorx Platform is located.

Typically, when available on devices which are compatible, the ThingWorx Edge MicroServer is installed on a client device for collecting and reporting data to ThingWorx using a secure WebSocket connection, instead of the Protocol Adapter SDK.

The Edge MicroServer is a powerful component of the ThingWorx architecture. The Edge MicroServer allows for the rapid deployment of connections between the ThingWorx Platform and an associated data reporting device, with minimal design requirements on the part of the user. It provides an "always- on" connection to the platform, and it opens a local web server that interacts with the REST API available on the platform.

About the ATrack AX7

The ATrack AX7 device is a vehicle diagnostics technology which provides real-time engine monitoring using GPS technology and CDMA/UMTS/HSPA mobile communication technology. The engine diagnostics data is collected through OBD-II communication port of the device and transmitted along with the other parameters such as GPS coordinates, the IMEI number of the device, an odometer reading, vehicle speed, and more via SMS/GPRS messages.

This guide will follow the features involved with setting up the device, simulating a connection, and pushing and pulling data from the device to a remote server.

Prerequisites

This guide assumes you have access to a server that is capable of accepting data over the internet from the ATrack AX7.

Although it is not required, the Protocol Adapter SDK for the ATrack AX7, included in this starter kit, should be run on the same server/local machine on which ThingWorx platform is run.

Please note that this guide was written for a computer running Microsoft Windows.

Refer to the Troubleshooting section of this guide for help with frequently asked questions.

It is recommended that you use a terminal tool such as Tera Term for serial communication purposes.

Configuration and Setup

Configuring the ATrack AX7

1. Power ON the device and ensure that the device has
   GSM and GPRS capability.
2. Setup and connect the AX7 device as mentioned in the Atrack AX7 User Manual.
3. Type and send the following commands to the device to configure it. For the purposes of this integration, the acknowledgment is disabled and the message format for the position report is set to ASCII.

Note that you will need to configure the *config.json* file in the AX7 source code to match the parameters you will configure the ATrack AX7 to connect to.

AT$FORM=0,,1

AT$TRAC=1,10,,,,,2

The **FORM** command sets the position report message format to ASCII.
The default message format is binary.

The **TRAC** command commences basic time mode tracking.
A position report is sent every 10 seconds.

4. Configure the AX7 device to send GPRS messages on the desired Host IP address and Host port. Enter the next command below, but replace the parameters in <> with their correct values for your use case. For example if the packet type is TCP, your command will look like this:
AT$GPRS=1,"neo.iot.net","","","52.11.96.167",5000,0,3,30,1,0
The IP address is the address of your server which will host the Protocol Adapter SDK, and the port is the port number it will listen on for data from the ATrack AX7 device. Take note of the values you use.

   AT$GPRS=<Enable>,<"APN">,<"Username">,<"Password">,<"Host Address">,<Host Port>,<Socket Type>,<Retry>,<Timeout>,<Keep Alive>,<Report ACK>, <Secondary Host IP Address/Domain name">,<UDP Local Port>

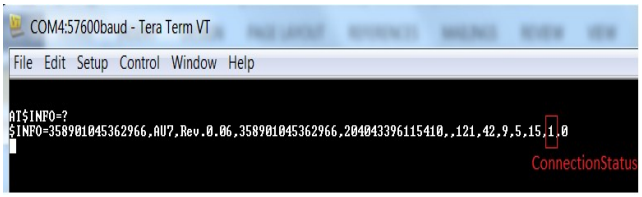
   Note: This command takes 20-30 seconds to complete, and then the device will reboot. Do not send any more commands during this period, until you see the "Bootloader..." message in the terminal. Send this command again after reboot:

   AT$TRAC=1,10,,,,,2

   Note: The above line will have to be entered every time the ATrack AX7 is rebooted.

   You should see a message such as "$OK" returned by the device after every command line sent.

5. Check if your device is connected by sending: AT$INFO=?
   You should receive something like the image shown at the right. The second-to-last parameter indicates the connection status. If it is "1", it means that the device is connected to your server.
   If you have not yet modified the *config.json* file so that the Protocol Adapter SDK listens on the same port you configured the AX7 to send to, (next step),
   you should see only a 0 at this point.



Note: All configurations can be tested by putting a '?' after the '=' sign. For example: AT$GPRS=?

Configuring the Protocol Adapter SDK

1. Locate and open the *config.json* file included in this Starter Kit and modify the socketPort setting in ConnectionServerSettings to "5000", or your chosen port.

   The Protocol Adapter SDK will be listening on this port. Remember that the GPRS port for the AX7 device, configured earlier, should be the same.

2. Set the following values:
   - Set thingworxServerAPIKey to the Application Key for your ThingWorx Platform. (Application keys can be generated directly on the Platform.)
   - Set thingworxServer and thingworxServerPort to the IP address and port number, respectively, of the machine that is running the ThingWorx Platform.

     (If the ThingWorx server and Protocol Adapter SDK are located on the same server, enter "localhost" for the thingworxServer parameter.)

   Your configuration should resemble the example on the right.

```
  "rows": [{
    "thingWorxServerUseSSL": false,
    "thingWorxServerAPIKey": "39d7be99-3631-4984-b891-6a1c57a4d824",
    "thingWorxServerTimeout": 60000,
    "thingWorxServer": "localhost",
    "thingWorxServerPort": 80
  }]
},
"ConnectionServerSettings": {
  "description": "Connection server / socket settings",
  "isMultiRow": false,
  "name": "ConnectionServerSettings",
  "ordinal": 0,
  "dataShape": {"fieldDefinitions": {
    "socketPort": {
      "baseType": "INTEGER",
      "description": "Port the server should listen on for incoming connecti
      "name": "socketPort",
      "aspects": {"defaultValue": 4444},
      "ordinal": 0
    },
    "connectionServerThingName": {
      "baseType": "STRING",
      "description": "Port to listen on",
      "name": "connectionServerThingName",
      "aspects": {"defaultValue": "ConnectionServer"},
      "ordinal": 0
    }
  }},
  "rows": [{
    "socketPort": 5000,
```

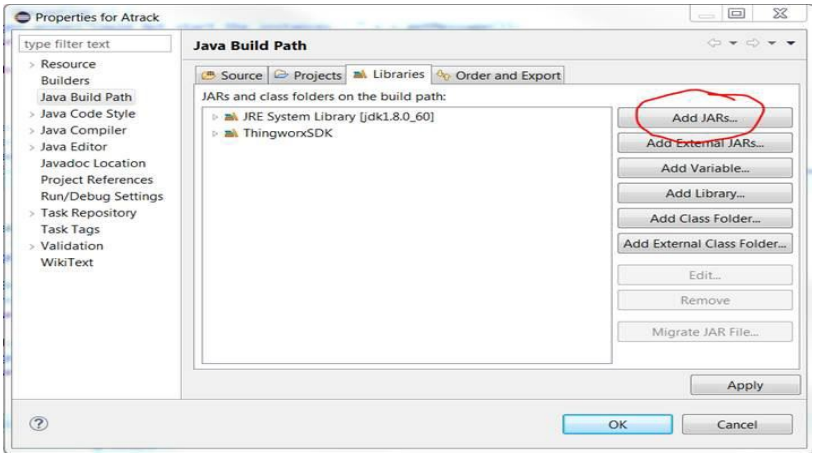
Running the Protocol Adapter SDK

Before running the Protocol Adapter SDK, make sure that your ThingWorx server is running on the configured port and address. Make sure that the ATrack device is plugged in, switched ON and it is kept in an area with high GPS visibility (such as near a window or outside).

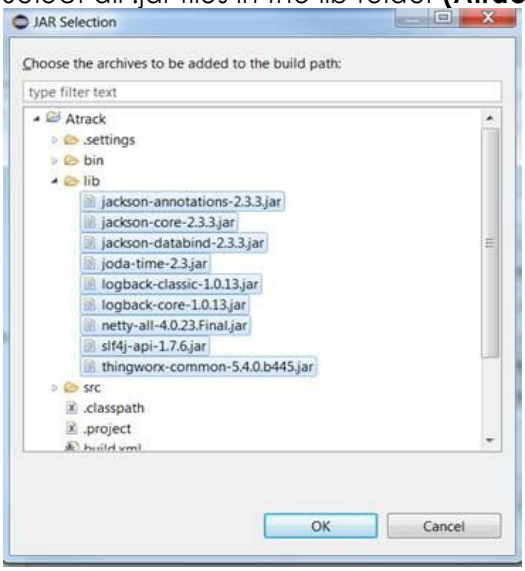
You may use an integrated IDE (such as Eclipse) to run the Protocol Adapter SDK, or you may compile the source into a jar file and run it through the command prompt.

This procedure assumes you are using Eclipse IDE.

1. Import the project into your Eclipse workspace (**File-> import-> Java-> Existing Projects...** ->), browse to the folder containing the project, and select **Copy into workspace**.
2. Configure the build path by adding the .jar files stored in the lib folder.
   (Right click on your Project -> Build path -> Configure build path).
   Select Add JARs in the Libraries tab.



Select all .jar files in the lib folder **(Atrack->lib)**

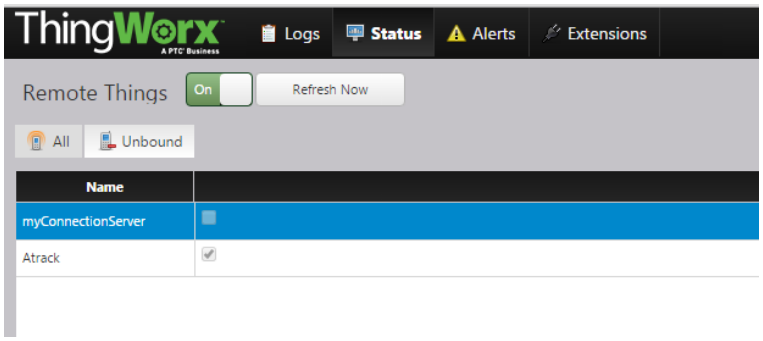


3. Using the Runnable jar: In a command prompt window, type the following command:
   Java –jar <jar-filename>
4. Run the ConnectionServer.java file in Eclipse, or the compiled JAR.
   You should see a log in the console that will give you clues as to the current status of the Protocol Adapter's connection to ThingWorx Server, and its current status in receiving incoming connections from the ATrack AX7.
   Refer to the troubleshooting section of this document for help.

When the ATrack AX7 has successfully connected to your Protocol Adapter SDK, you should see a log in the console similar to the following as it receives and forwards data to the ThingWorx server:

```
Message Processed!
Properties updated!
19:59:08.156 [Thread-9] INFO  c.t.d.server.SocketHandler - Incoming message:20150924185723,20150924185723,20150925195904,-75672250,40055820,322,2,244,75,0,17,0,0,,2000,2000,□
Handle Ingest message
Begin Parsing!
Longitude: -75.67225
Latitude: 40.05582
Message generated!
Message Processed!
Properties updated!
19:59:08.156 [Thread-9] INFO  c.t.d.server.SocketHandler - Incoming message:20150924185732,20150924185733,20150925195904,-75672671,40055550,267,2,245,75,0,7,0,0,,2000,2000,□
Handle Ingest message
Begin Parsing!
Longitude: -75.672671
Latitude: 40.05555
Message generated!
Message Processed!
Properties updated!
19:59:08.156 [Thread-9] INFO  c.t.d.server.SocketHandler - Incoming message:20150924185743,20150924185743,20150925195904,-75672848,40055380,90,2,248,75,0,35,0,0,,2000,2000,□
Handle Ingest message
Begin Parsing!
Longitude: -75.672848
Latitude: 40.05538
Message generated!
Message Processed!
Properties updated!
```

5. Browse to your ThingWorx Platform.
6. Navigate to the **Monitoring** > **Remote Things** section and verify that you see two unbound things: *myConnectionServer* (may be different depending on the configuration in your *config.json* file), and *ATrack*.

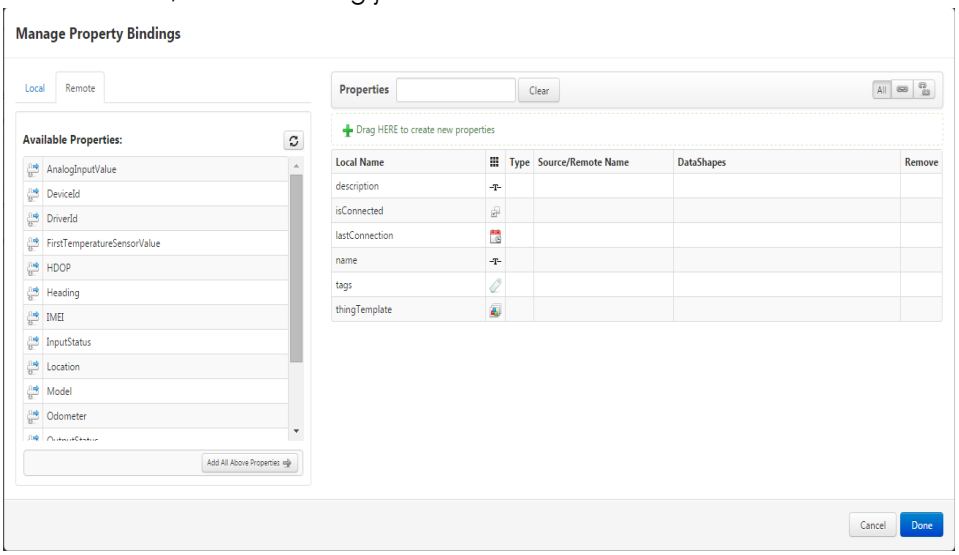


7. Create a new Thing entity in ThingWorx which implements the Remote Thing template.

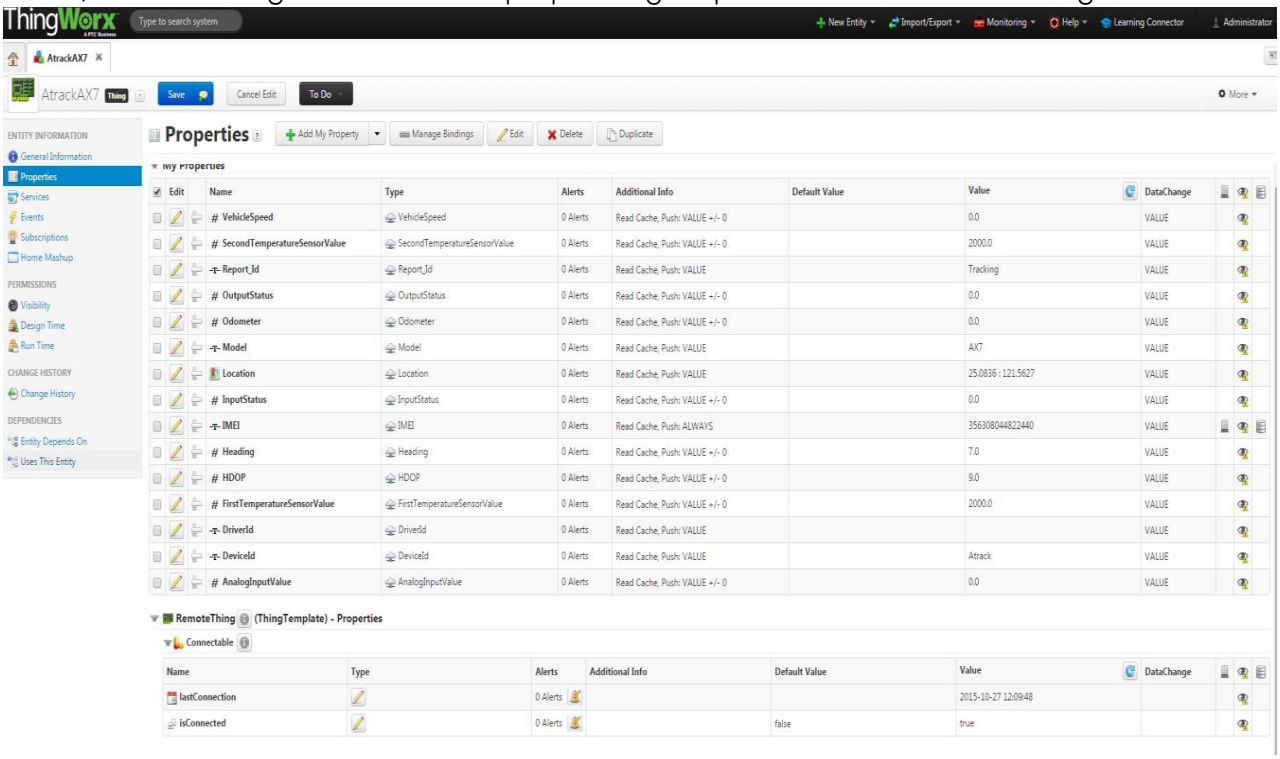
   For the Thing's "Identifier" property (General Information page), click **Browse** and choose "ATrack" which should be visible as an identifier.

   Save the Thing.

8. Now, click **Properties** - **Manage Bindings** - **Remote Bindings**. A list of properties for the Atrack AX7 appears.
   The default list (shown in the following image) here may be modified in the Protocol Adapter SDK source code, in *AtrackThing.java*.



Select "Add All Above Properties" to add all the remote properties to the AtrackThing. Hit Done, save the thing and refresh. The properties get updated as shown in the image below:



This completes the configuration of your ATrack AX7 device and its connection to ThingWorx Platform.

Troubleshooting

	Problem	Solution(s)
1	Protocol Adapter is reporting this debug message: [ClientHandler: 1299411443] Client has been idle, sending websocket ping...	This is a normal communication when the Protocol Adapter is receiving no data from the ATrack device. 1. Verify that *config.json* is configured with the correct Thingworx application key and it is set to receive messages on the desired host port and IP address. 2. Check the configuration and reset the device by sending the configuration AT$ commands mentioned earlier in this guide. Wait for the device to reboot after changing the server connection information. Make sure that the GPRS server hostname is set to the "public" IP address of the Computer/AWS server running the protocol adapter SDK. 3. Disable the firewall setting for TCP port 5000 (or the port you are using for your connection, as defined in *config.json*). This is the socket port the Protocol Adapter SDK will listen on for incoming ATrack AX7 messages. 4. Ensure that the ATrack AX7 device has GSM signal (check the status LED on the front). 5. Check the AT$TRAC command once again by sending "AT$TRAC=?" on the terminal via Serial connection. If you do not get 1,10,,,,,2 as the response, then send the AT$TRAC command again to set it. Ensure that the device is disabled for acknowledgment and the periodic report time is set to 10 seconds.
2	The ConnectionServer.java cannot connect to the ThingWorx platform.	1. Verify that the parameters in *config.json* are correct; especially the Application Key, the server address, and the connection port. 2. Verify that *config.json* is in the root directory of the ATrack SDK. 3. Verify that no firewalls are blocking your connection to the Platform. 4. Verify that the port chosen for your application is not closed on your server.

Compatibility

This guide has been tested for compatibility with the DEVICE and the following ThingWorx Platform and operating system:

ThingWorx Platform Version	ThingWorx 6.0.1
OS	Windows 7, Service Pack 1, Windows Server 12 hosted on AWS

Document Revision History

Revision Date	Version	Description of Change
October 27, 2015	1.0	Initial Release